

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-035883

(43)Date of publication of application : 12.02.1993

(51)Int.Cl.

G06F 15/72
G06F 3/153
G06F 15/72
G09G 5/36

(21)Application number : 03-189795

(71)Applicant : MATSUSHITA ELECTRIC IND CO
LTD

(22)Date of filing : 30.07.1991

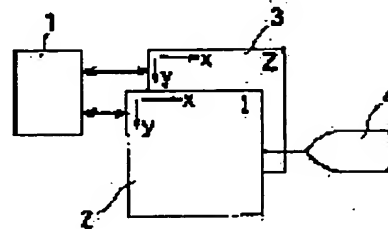
(72)Inventor : MINEHISA JIRO

(54) THREE-DIMENSIONAL IMAGE GENERATING METHOD

(57)Abstract:

PURPOSE: To obtain the three-dimensional image generating method which can eliminate processing having no direct contribution to image generation.

CONSTITUTION: A processor 1 compares a depth coordinate value (z) among screen coordinates with the value in a Z buffer memory 3 in an implicit surface erasure processing step based upon Z buffer algorithm to decide the front-rear relation of a figure in picture element units and select graphic data which are closer to a point of vision, and stores the identification number of the figure in a frame buffer 2. Further, the processor 1 calculates brightness I on a screen in the picture element units in a brightness calculation step by using the identification number in the frame buffer 2 at the end of the implicit surface erasure processing step for all graphic data and stores the brightness in the frame buffer memory 2. Then the processor 1 displays a three-dimensional image on a display device 4 according to the contents of the frame buffer memory 2.



(19) 日本国特許庁(JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平 5-35883

(43)公開日 平成5年(1993)2月12日

(51) Int. C1. 5		識別記号	庁内整理番号	F I	技術表示箇所
G O 6 F	15/72	4 2 0	9192- 5 L		
	3/153	3 2 0 N	9188- 5 B		
	15/72	4 6 5	9192- 5 L		
G O 9 G	5/36		8121- 5 G		

審査請求・未請求 請求項の数 6

(全15頁)

(21)出願番号 特願平3-189795

(22) 出願日 …… 平成3年(1991)7月30日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 發明者 峰久 次郎

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

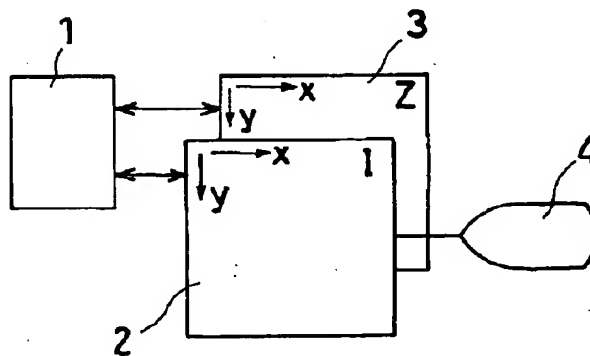
(74) 代理人 弁理士 中島 司朗

(54) 【発明の名称】 三次元画像生成方法

(57) 【要約】

【目的】画像生成に直接的に寄与しない処理を削減できる三次元画像生成方法を提供する。

【構成】プロセッサ 1 は、Z バッファ・アルゴリズムに基づいた隠面消去処理ステップで、スクリーン座標のうちの奥行き座標値 z を Z バッファ・メモリ 3 の値と比較することにより、各画素毎に図形の前後を判定して視点により近い図形データを選択し、その図形の識別番号をフレーム・バッファ・メモリ 2 に格納する。さらにプロセッサ 1 は、輝度算出ステップで、全ての図形データに対する隠面消去処理ステップの終了後に、フレーム・バッファ・メモリ 2 の識別番号を用いてスクリーン上の各画素毎に輝度 I を算出し、フレーム・バッファ・メモリ 2 に格納する。そしてプロセッサ 1 は、フレーム・バッファ・メモリ 2 の内容に基づいて三次元画像を表示装置 4 に表示させる。



るいは「基礎グラフィックス」川合慧著、昭晃堂、1985年、186ページなどに記載されている。まずフレーム・バッファ・メモリ32およびZバッファ・メモリ33を初期化する(ステップS101)。すなわち、フレーム・バッファ・メモリ32には背景色の輝度を、Zバッファ・メモリ33には奥行き座標値Zの最大値(最も視点から離れた位置の座標値)を設定する。次にポリゴンを任意の順序でスキャン変換する(ステップS102)。次にスキャン変換したポリゴン内の各画素に対して奥行き座標値Zを計算する(ステップS103)。次に算出した奥行き座標値ZとZバッファ・メモリ33に格納されている奥行き座標値ZBとを比較し(ステップS104)、算出した奥行き座標値Zがフレーム・バッファ・メモリ32に格納されている奥行き座標値ZBよりも小さい場合、すなわち視点により近い位置にある場合、Zバッファ・メモリ33の値を算出した奥行き座標値Zに置き換え(ステップS105)、シェーディング計算(ステップS106)から当該画素の輝度を求めてフレーム・バッファ・メモリ32に格納する(ステップS107)。ステップS104において、ステップS103で算出した奥行き座標値ZがZバッファ・メモリ33に格納されている奥行き座標値ZB以上であると判断した場合、すなわち視点から遠い位置にある場合、あるいはステップS107の処理を実行した後、ポリゴンに含まれるすべての画素に対してステップS103～S107の処理が終了したか否かを判断し(ステップS108)、終了していなければステップS103に戻る。終了していれば、全てのポリゴンに対してステップS102～S108の処理が終了したか否かを判断し(ステップS109)、終了していなければステップS102に戻る。終了していればこのルーチンを終了する。

【0004】以上の処理が終了した時点では、フレーム・バッファ・メモリ32にはスクリーン上の各画素について視線に最も近いポリゴンの輝度だけが格納されており、求める三次元画像が生成されている。特に、滑らかな陰影付け(シェーディング)を行うスムーズ・シェーディングの1つのアルゴリズムであるグロー・シェーディング・アルゴリズムに基づく計算においては、奥行き座標値Zと同様にして輝度Iを逐次算出することができ、より高速に三次元画像を生成できる。

【0005】

【発明が解決しようとする課題】しかしながら上記従来の方法では、隠面消去が終了する前にシェーディング計算(輝度算出)が行われるので、より視点に近いポリゴンが後から処理された場合にはシェーディング計算をやり直す必要があり、最終結果の画像生成に寄与しない処理の時間が増加するという問題を有していた。特に、グロー・シェーディングよりも高画質の三次元画像を生成するためには、より複雑なシェーディング計算を行う必要があり、取り扱う形状データの量が増加する。すな

わち、より高画質の三次元画像を追求するほど、無効になるシェーディング計算が増加し、最終結果の画像生成に寄与しない処理の時間が増加してしまう。

【0006】本発明はかかる事情に鑑みて成されたものであり、画像生成に直接的に寄与しない処理を削減できる三次元画像生成方法を提供することを目的とする。

【0007】

【課題を解決するための手段】請求項1の発明は、スクリーン座標のうちの奥行き座標値zを用いて各画素毎に図形の前後を判定して視点により近い図形データを選択するZバッファ・アルゴリズムに基づいた隠面消去処理ステップと、全ての図形データに対する前記隠面消去処理ステップの終了後にスクリーン上の各画素毎に輝度Iを算出する輝度算出ステップとを実行することを特徴としている。

【0008】請求項2の発明は、請求項1の発明において、図形の輝度Iが、スクリーン座標のうちの(x, y)座標値により一意に決定できる、すなわち関数gを用いて $I = g(x, y)$ として定義できる場合に、輝度算出ステップにおいて、輝度Iを画素の位置すなわち(x, y)座標値より式 $I = g(x, y)$ を用いて直接的に算出することを特徴としている。

【0009】請求項3の発明は、請求項1の発明において、図形の輝度Iが、スクリーン座標のうちの(x, y)座標値により一意に決定できる、すなわち関数gを用いて $I = g(x, y)$ として定義でき、かつ直前に輝度値を算出した画素における輝度I'と関数gのx座標またはy座標に対する偏微分成分 $g'(x, y)$ とを用いて $I = I' + g'(x, y)$ と算出できる場合に、輝度算出ステップにおいて、輝度Iを算出すべき画素の図形が直前に輝度I'を算出した画素の図形と同じときには輝度Iを前記式 $I = I' + g'(x, y)$ を用いて算出し、輝度Iを算出すべき画素の図形が直前に輝度値を算出した画素の図形と異なるときには輝度Iを前記式 $I = g(x, y)$ を用いて直接的に算出することを特徴としている。

【0010】請求項4の発明は、二次元のスクリーン平面上の座標(x, y)と前記スクリーン平面からの奥行きを表わす座標zとからなるスクリーン座標(x, y, z)により定義される三次元の図形データと、色や輝度や法線ベクトルなどの図形特有の性質を決定する属性データの組からなる属性ベクトル(t_1, t_2, \dots, t_n) (nは2以上の整数)とを用いて立体的な画像を生成する三次元画像生成方法において、前記スクリーン座標のうちの奥行き座標値zを用いて各画素毎に図形の前後を判定して視点により近い図形データを選択するZバッファ・アルゴリズムに基づいた隠面消去処理ステップと、全ての図形データに対する前記隠面消去処理ステップの終了後にスクリーン上の各画素毎に属性ベクトルの要素である属性要素 t_i (iは1からnまでの整数)の

ム・バッファ・メモリ2とZバッファ・メモリ3とを初期化する(ステップS1)。すなわち、フレーム・バッファ・メモリ2にはポリゴンの識別番号(以下「ポリゴンid」と記す)として0を、Zバッファ・メモリ3には奥行き座標値Zの最大値(最も視点から離れた位置の座標値)を設定する。次にポリゴンidの順序でポリゴンをスキャン変換する(ステップS2)。次にポリゴン内の各画素に対して奥行き座標値Zを計算する(ステップS3)。次に算出した奥行き座標値ZとZバッファ・メモリ3に格納されている奥行き座標値ZBとを比較し(ステップS4)、算出した奥行き座標値ZがZバッファ・メモリ3に格納されている奥行き座標値ZBよりも小さい場合、すなわち視点により近い位置にある場合、Zバッファ・メモリ3の値を算出した奥行き座標値Zに置き換え(ステップS5)、フレーム・バッファ・メモリ2に当該ポリゴンのポリゴンidを格納する。(ステップS6)。ステップS4において、ステップS3で算出した奥行き座標値ZがZバッファ・メモリ3に格納されている奥行き座標値ZB以上であると判断した場合、すなわち視点から遠い位置にある場合、あるいはステップS6の処理の終了後、ポリゴンに含まれるすべての画素に対してステップS3～S6の処理が終了したか否かを判断し(ステップS7)、終了していなければステップS3に戻る。終了していれば、全てのポリゴンに対してステップS2～S7の隠面消去処理が処理が終了したか否かを判断し(ステップS8)、終了していなければステップS2に戻る。終了していれば、各画素毎にシェーディング計算(輝度算出)を行ってフレーム・バッファ・メモリ2に格納し(ステップS9)、このルーチンを終了する。

【0020】図3はシェーディング計算(輝度算出)の手順を説明するフローチャートで、これは図2のステップS9に相当する。なお、スクリーン上の座標(x, y)は、x, y共に正の整数値のみで表されるものとする。まず処理対象とする画素の位置を示すスクリーン上の座標(x, y)のうちのy座標を1に初期化する(ステップS11)。次に同じくスクリーン上の座標(x, y)のうちのx座標を1に初期化する(ステップS12)。次にスクリーン上の座標(x, y)でのポリゴンidをフレーム・バッファ・メモリ2から読み出す(ステップS13)。次にポリゴンidで指定される輝度算出のための関数gを選択する(ステップS14)。次に関数gとスクリーン上の座標(x, y)とを用いて輝度Iを式 $I = g(x, y)$ で算出する(ステップS15)。すなわちシェーディング計算(輝度算出)を行う。次に算出した輝度Iをフレーム・バッファ・メモリ2の(x, y)の位置に格納する(ステップS16)。次にx座標を1だけ増加させる(ステップS17)。次に新しいx座標の値とスクリーン上でのx座標の最大値Xmaxとを比較し(ステップS18)、新しいx座標

がXmax以下ならばステップS13へ戻る。新しいx座標がXmaxよりも大きいならば、y座標を1だけ増加させる(ステップS19)。次に新しいy座標の値とスクリーン上でのy座標の最大値Ymaxとを比較し(ステップS20)、新しいy座標がYmax以下ならばステップS12へ戻る。新しいy座標がYmaxよりも大きいならばこのルーチンを終了する。

【0021】このように、隠面消去処理後の最終画像に対応する画素のみに対してシェーディング計算を行うので、無効となるシェーディング計算を排除することができる。例えば $g(x, y)$ として(x, y)に関する1次式を用いると、グーロー・シェーディング・アルゴリズムに基づいた輝度を求めることができ、従来の方法と同画質の画像を短時間で得ることができる。取り扱うポリゴンデータが増加するに従い、本発明による処理数の低減効果がより顕著になることは明かである。

(実施例2) 実施例1では図3の手順によりシェーディング計算(輝度算出)を行ったが、図4のような手順でシェーディング計算(輝度算出)を行ってもよい。すなわち、まず処理対象とする画素の位置を示すスクリーン上の座標(x, y)のうちのy座標を1に初期化する(ステップS21)。次に同じくスクリーン上の座標(x, y)のうちのx座標を1に初期化する(ステップS22)。次にスクリーン上の座標(x, y)のポリゴンidをフレーム・バッファ・メモリ2から読み出す(ステップS23)。次に読み出したポリゴンidと隣接した画素のポリゴンidとを比較し(ステップS24)、読み出したポリゴンidと隣接画素のポリゴンidとが等しい場合、輝度Iを算出する関数 $g(x, y)$ のx方向の偏微分成分の画素(x, y)での値 g'

(x, y)を算出する(ステップS25)。次に画素(x, y)の隣接画素(x-1, y)における輝度 I' と $g'(x, y)$ とを用いて、画素(x, y)の輝度Iを式 $I = I' + g'(x, y)$ で算出する(ステップS26)。ステップS24において、ステップS23で読み出したポリゴンidと隣接画素のポリゴンidとが等しくないと判断した場合、読み出したポリゴンidで指定される輝度算出のための関数gを選択する(ステップS27)。次に関数gとスクリーン上の座標(x, y)とを用いて輝度Iを式 $I = g(x, y)$ で算出する(ステップS28)。次にステップS26あるいはステップS28で算出した輝度Iをフレーム・バッファ・メモリ2の(x, y)の位置に格納する(ステップS29)。次にx座標を1だけ増加させる(ステップS30)。次に新しいx座標の値とスクリーン上でのx座標の最大値Xmaxとを比較し(ステップS31)、新しいx座標がXmax以下ならばステップS23へ戻る。新しいx座標がXmaxよりも大きいならば、y座標を1だけ増加させる(ステップS32)。次に新しいy座標の値とスクリーン上でのy座標の最大値Ymaxとを比較し

期化する(ステップS61)。次に同じくスクリーン上の座標(x, y)のうちのx座標を1に初期化する(ステップS62)。次にスクリーン上の座標(x, y)のポリゴンidをフレーム・バッファ・メモリ12から読み出す(ステップS63)。次に読み出したポリゴンidと隣接する画素のポリゴンidとを比較し(ステップS64)、読み出したポリゴンidと隣接画素のポリゴンidとが等しい場合、属性要素tiを算出する関数gi(x, y)のx方向の偏微分成分の画素(x, y)での値g'(x, y)を算出する(ステップS65)。次に画素(x, y)の隣接画素(x-1, y)における属性要素の値t'iとg'i(x, y)とを用いて、画素(x, y)の属性要素tiを式 $t_i = t'_i + g'_i(x, y)$ で算出する(ステップS66)。ステップS64において、ステップS63で読み出したポリゴンidと隣接画素のポリゴンidとが等しくないと判断した場合、読み出したポリゴンidで指定される属性要素の値算出のための関数giを選択する(ステップS67)。次に関数giとスクリーン上の座標(x, y)とを用いて、属性要素tiを式 $t_i = g_i(x, y)$ で算出する(ステップS68)。ステップS66あるいはステップS68の処理を実行した後、算出した属性要素tiを属性バッファ・メモリ15の(x, y)の位置に格納する(ステップS69)。ステップS65〜S69の処理は必要な属性要素すべてに対して行う。次にx座標を1だけ増加させる(ステップS70)。次に新しいx座標の値とスクリーン上でのx座標の最大値Xmaxとを比較し(ステップS71)、新しいx座標がXmax以下ならばステップS63へ戻る。新しいx座標がXmaxよりも大きいならば、y座標を1だけ増加させる(ステップS72)。次に新しいy座標の値とスクリーン上でのy座標の最大値Ymaxとを比較し(ステップS73)、新しいy座標がYmax以下ならばステップS62へ戻る。新しいy座標がYmaxよりも大きいならばこのルーチンを終了する。

【0025】このように、関数g(x, y)のx方向の偏微分成分g'(x, y)を用いれば、演算が容易になり、処理速度を高速化できる。なお関数g(x, y)のy方向の偏微分成分を用いてもよい。

【0026】

【発明の効果】以上説明したように本発明によれば、全

ての図形データに対する隠面消去処理を行った後に、スクリーン上の各画素毎に直接的に輝度Iもしくは属性要素tiの値を算出するので、最終結果の画像生成に寄与しない部分に対する処理を削除し、より高画質な画像を効率よく生成することができるという優れた効果を奏する。

【図面の簡単な説明】

【図1】本発明の実施例1における三次元画像生成方法を実現する三次元画像生成装置のブロック図である。

【図2】本発明の実施例1における三次元画像生成方法の手順を説明するフローチャートである。

【図3】本発明の実施例1における三次元画像生成方法で採用した輝度算出手順を説明するフローチャートである。

【図4】本発明の実施例2における三次元画像生成方法で採用した輝度算出手順を説明するフローチャートである。

【図5】本発明の実施例3における三次元画像生成方法を実現する三次元画像生成装置のブロック図である。

【図6】本発明の実施例3における三次元画像生成方法の手順を説明するフローチャートである。

【図7】本発明の実施例3における三次元画像生成方法で採用した属性値算出手順を説明するフローチャートである。

【図8】本発明の実施例4における三次元画像生成方法で採用した属性値算出手順を説明するフローチャートである。

【図9】従来の三次元画像生成方法を実現する三次元画像生成装置のブロック図である。

【図10】従来の三次元画像生成方法の手順を説明するフローチャートである。

【符号の説明】

1 プロセッサ

2 フレーム・バッファ・メモリ

3 Zバッファ・メモリ

4 表示装置

11 プロセッサ

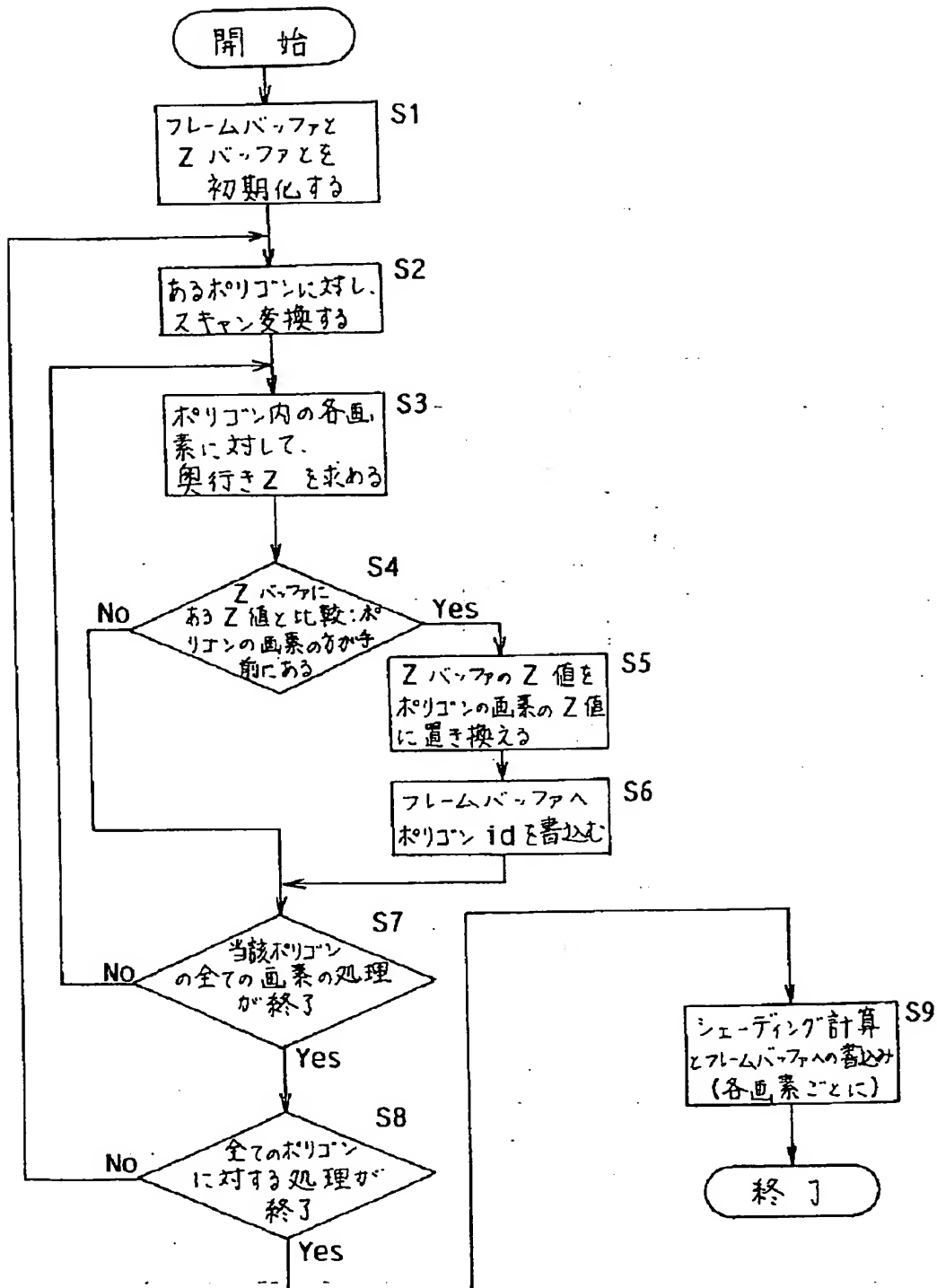
12 フレーム・バッファ・メモリ

13 Zバッファ・メモリ

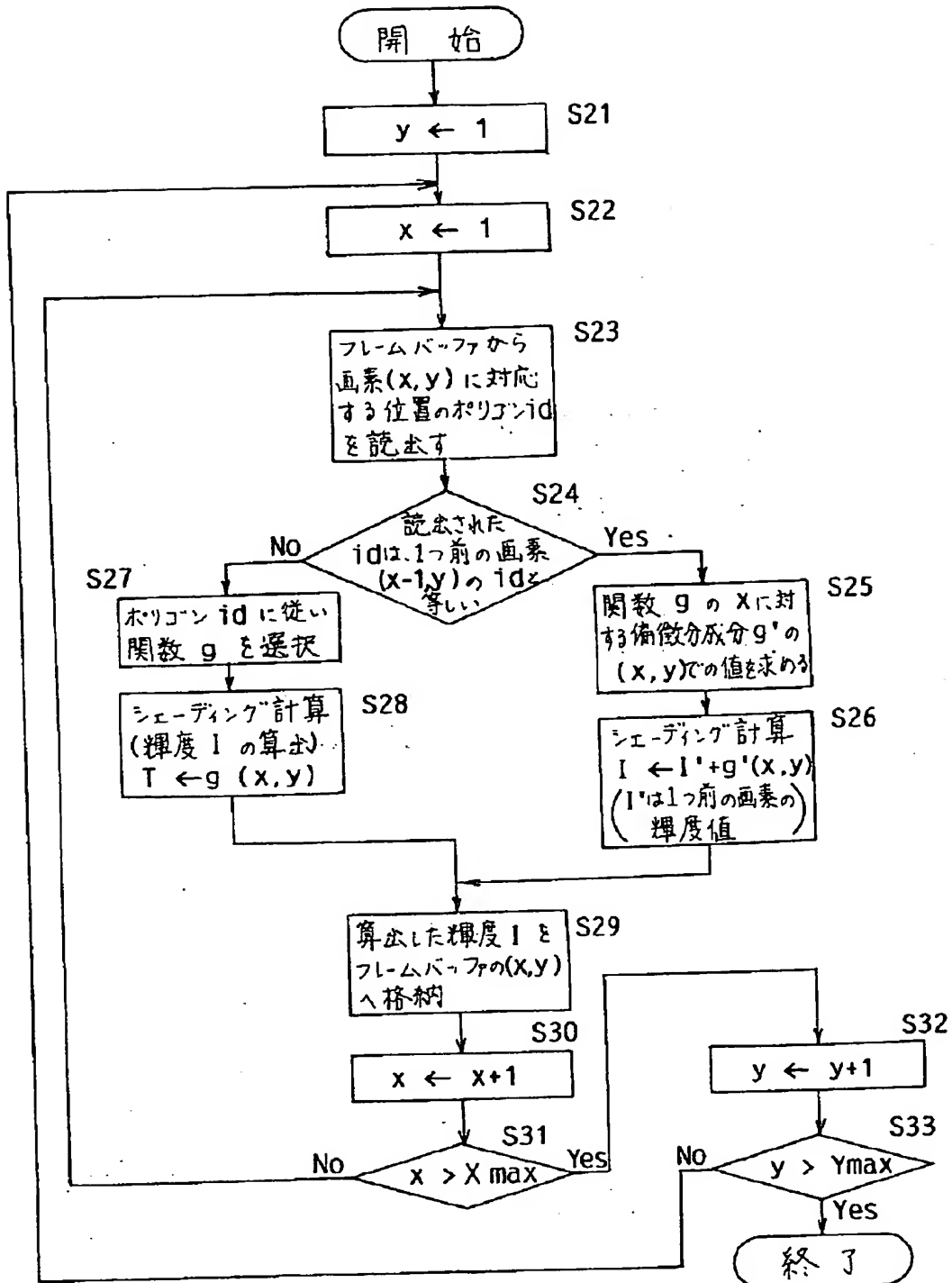
14 表示装置

15 属性バッファ・メモリ

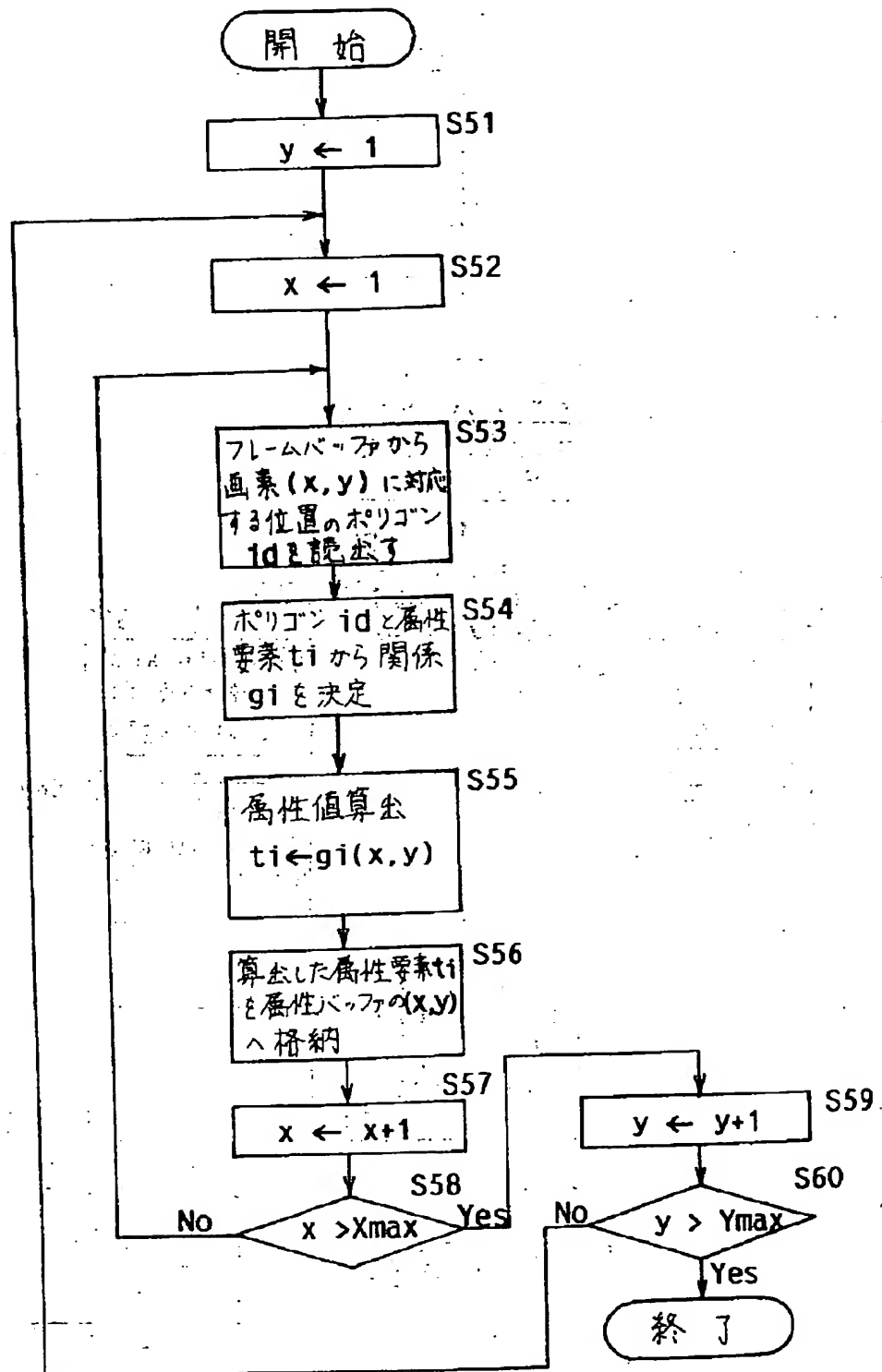
【図 2】



【図4】



【図 7】



【図10】

